eCl@ss-BMEcat-Guideline
proposal for an embedding of eCl@ss into BMEcat

Version <1.0>

Revision History

Date	Version	Description	Author
15.11.2011	1.0	initial version	eCl@ss CRD

eCl@ss center of research and development (CRD):

Harald Beckmann, ProPlan-Consulting
Dr. Wilfried Hartmann, BASF SE
Gerd Koziel, Siemens Energy
Gerald Lobermeier, Weidmüller Interface GmbH & Co. KG
Stefan Mülhens, AmpereSoft GmbH
Nikolaus Ondracek, Paradine GmbH
Mathias Richter, Paradine GmbH
Josef Schmelter, PHOENIX CONTACT GmbH & Co. KG
Frank Scherenschlich, Class.Ing Ingenieur-Partnerschaft
Henning Uiterwyk, eCl@ss

and
Felix Hettig, eCl@ss
Dr. Matthias Richter, Paradine GmbH

Please send remarks to crd@eclass.de

Table of Contents

1 Introduction

1.1 Purpose

In this document the embedding and usage of eCl@ss content in the context of BMEcat catalogs is described as a basis for discussion with BME and as a guideline to implementers.

1.2 Scope

In scope is:

- A guideline for the embedding of eCl@ss in general into BMEcat2005, not limited but with particular regard to eCl@ss 7 and BMEcat 2005.1

- Transaction type is T_NEW_CATALOG

Out of scope is:

- BMEcat 1.2 (or earlier)

- UDX (user defined extensions)

- Other feature group systems, be they external or transported in the catalog

1.3 Definitions, Acronyms and Abbreviations

Catalog

> document containing product data

Catalog requirements

> formal expression of needs about the properties used to describe product data

Coded name

> code to describe an eCl@ss class
>
> It is built up in 8 digits describing the 4 levels defined in eCl@ss

Classification

> a (complete, mutually exclusive) segmentation of the world according to consistent, unique principles

Classification (process)

> assignment of labels to collections of objects based on rules about common characteristics

Classification (system)

> (hierarchical) taxonomy of class labels and rules for their assignment

(Data) dictionary

collection of (product) models and classification system(s)

Dictionary change management

process and rules governing the evolution and usage of data dictionaries

Multi-classification

a) assignment of labels of multiple classification systems to objects

Example: According to eCl@ss "peanuts" are classified as: 16040402, according to CPV they are classified as 03111200-4

b) assignment of multiple labels from one classification system to objects

Special case: in classifications with an is_a hierarchy (e.g. eCl@ss), lower level classes are always (specialized) higher level classes, e.g. peanut is a nut, is a fruit, is a food.

Note: In monothetic, hierarchical classification systems such as eCl@ss only the more generic levels could occur in multiclassification. In case of eCl@ss they are implicitly contained in the coded name.

Note 2: In polythetic classification systems the number of class labels of objects are related to the applicable classification aspects of the system.

Multi-valuation

expression that *any or all* of the elements of the set of values apply for a certain property

Representation

formal description of certain objects and their characteristics

Template

a template is a specification of the format and data requirements for a catalogue to meet the needs of a data recipient (e.g. a buyer)

1.4 Basic assumptions

The following assumptions are made in this document:

- The eCl@ss dictionary is available separately from the BMEcat catalog, i.e. eCl@ss is not to be transported inside the catalog

- There is a process in place that guarantees that **catalog requirements** can be expressed in a machine tractable way

- Concept identifiers are used to identify elements from the dictionary

- The dictionary applies dictionary change management

1.5 References

eCl@ss Release

More information on eCl@ss can be found at: www.eclass.de

BMEcat

More information on BMEcat can be found at: www.bmecat.org

Advice to BMEcat 2005.1 document and eCl@ss Wiki (examples)

ISO 29002-5:

Industrial automation systems and integration -- Exchange of characteristic data -- Part 5: Identification scheme

2 Prerequisites

2.1 eCl@ss

2.1.1 Identification

2.1.1.1 Concept identifier (ISO 29002-5)

For referencing eCl@ss classes and properties, etc. the elements have to be identified. As catalogues are exchanged worldwide and eCl@ss is available in different languages the identification of the elements (classes, properties, etc.) is NOT done via language dependent names. For identifying eCl@ss elements (class, property) the identification schema according to ISO 29005-5 is used.

The picture below shows the detailed structure of the concept identifier according to ISO 29005-5.

Figure 1: Identification schema according to ISO 29005-5
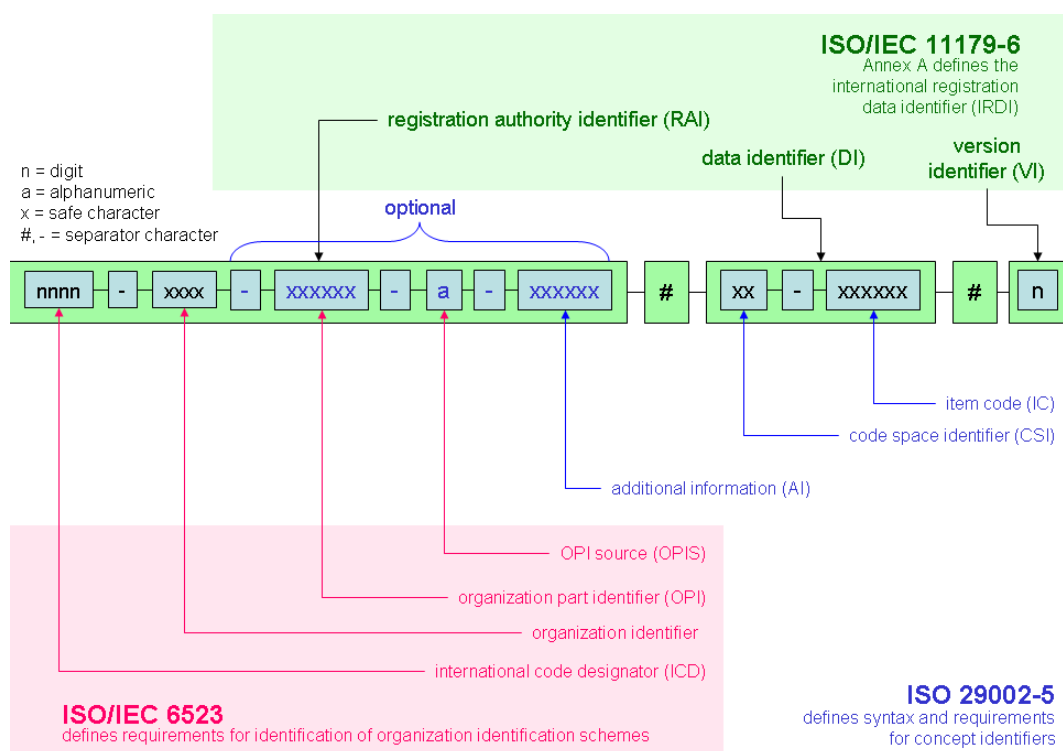
The Code Space Identifier (CSI) defines the category of the item. The table below shows an excerpt of CSIs used for eCl@ss elements.

Code Space Identifier (CSI)	Category of administrated item
01	class
02	property
05	unit of measurement
07	property value

Table 1: Excerpt of Code Space Identifiers (CSI) according to ISO 29005-5

The below table shows an example of an identification code for an eCl@ss class. It becomes clear that the length of this IRDI is well below the 60 character limit for the length of an identifier in the current BMEcat format. So it can be assumed that it's save to write an IRDI as an eCl@ss identifier into the catalogue.

0173-1#01-AAA123#001	
Code:	description
0173	ICD code for eCl@ss
1	eCl@ss Office
01	class
AAA123	identifier of class
001	version of class

Table 2: Identification code for an eCl@ss class according to ISO 29005-5

2.1.1.2 eCl@ss Coded Name

The hierarchy of the classification classes is represented with the help of the **coded name**, i.e. the class code. The coded name consists of an 8-digit integer number, two digits for each hierarchical level. The number of trailing zeros in the end indicates the level of hierarchy, e.g. 16000000 (Segment "Food, beverage, tobacco"), 16040000 (Main group "Fruit"), 16040300 (Group "Berry fruit"), 16040301 (Commodity class "Blackberry"). The fourth level, the commodity class or product group is then further described with the help of properties and property values. Properties and values form the basis for the product description.

eCl@ss class coded name has to be written as follows:

[Segment][Main Group][Group][Commodity Class]

Example: 27010110

Note 1:
The coded name is an identification of the classification and is complementary to the concept identifier (see 2.1.1.1).
Note 2:
Dashes are not allowed.
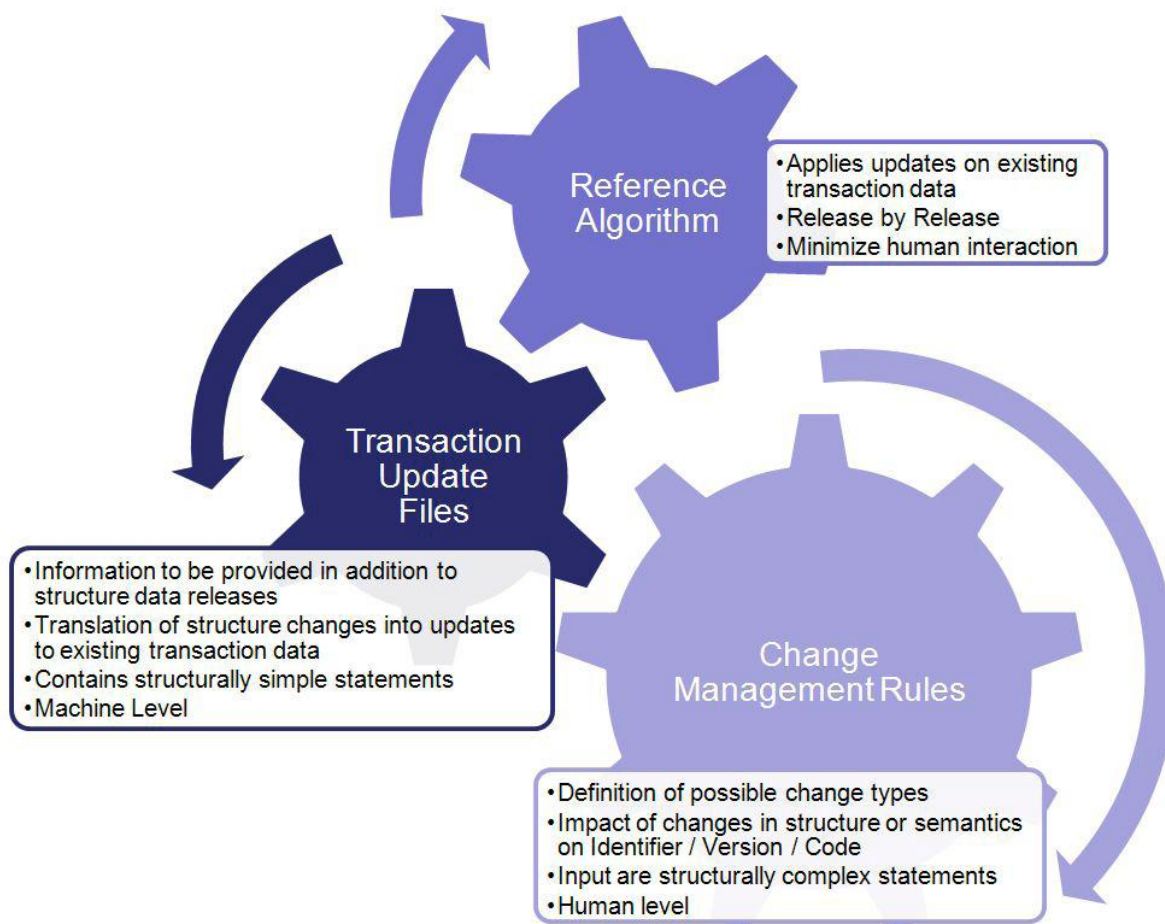
2.1.2 Dictionary change management

Figure 2: Interdependencies in dictionary Change Management

Data provider and user side of a dictionary focus on different types of data. Structure data is maintained by the registration authority that has command over content and identifiers. Transaction data and templates created in reference to certain structure data releases but otherwise independent. Dictionary Change Management has the purpose of relating these perspectives. Therefore changes to the dictionary are

> (1) transcribed into defined modifications of structure data such that
> (2) update paths for existing transaction data and views are provided under all circumstances
> (3) and efforts on user side for applying these updates are kept minimal

This is in line with the scopes of the principal components of dictionary change management:

> (1) Change Management Rules
> (2) Transaction Update Files
> (3) Reference Algorithm

From the above description it becomes clear that any of the components can only make sense in interaction with the other two, as illustrated in Figure 2.

The matter of change management rules is complex operations performed on structure elements, their attributes and units seen in the context of three principal concepts that impact change:

1. Identity stands behind the notion of semantic change that is taken as the decision criterion for assigning revisions, versions or new identifiers. Some of these distinctions have to be made explicit in change requests.
2. Obligation influences whether something, especially an attribute of a property, class or unit, may be added, deleted or only modified. The concept of obligation is represented only in the data model and not in the structure data. Thus it cannot be a target for change requests.

3. Dependency deals with the non-local impacts of changes, most notably in the case that complex structure elements are involved. Dependencies can and should be resolved automatically by the portal system based on the choices made in 1. As a result more structure elements get updated automatically.

The matter of transaction update files is simple, atomic operations performed on valuations generated in reference to a certain structure data release. In principle transaction data stores valuations as values that can be accessed by a path that allows the allocation of the value to the correct structure element, attribute and unit. Transaction update files are the defined way of converting views as well as paths and values in transaction data between subsequent structure data releases.

The matter of the reference algorithm is entries in the transaction update file and their preferably automatic application on transaction data. Applying the reference algorithm requires access to the appropriate releases of the structure data as only such entries are contained in the transaction update files that cannot be retrieved from comparing the releases of the structure data directly.

2.1.3 Two representation

2.1.3.1 Basic representation

Until eCl@ss 7.0 the eCl@ss dictionary only contained the basic representation (without explicitly saying so). Catalogs with product descriptions according to the basic representation could use both, the BMEcat2005 format and BMEcat2005+onto, depending on the underlying representation (see Section 2.1.4).

2.1.3.2 Advanced representation

Starting with eCl@ss 7.0, for some classes an advanced representation was introduced in addition to the basic representation. There exists a complete mapping of properties in the context of the basic representation into their corresponding context in the advanced representation.

Product descriptions according to the advanced model require the BMEcat 2005.1 format.

2.1.4 Deliveries of the eCl@ss dictionary

Starting with eCl@ss 7.0, the eCl@ss dictionary is delivered in three formats, one CSV based and two XML based formats.

2.1.4.1 eCl@ss basic CSV

The eCl@ss basic CSV format contains only the basic representation and does not take into consideration of the default aspect. Therefore catalogs based on this delivery can be created in the BMEcat2005 format.

2.1.4.2 eCl@ss basic XML

The eCl@ss basic XML format contains only the basic representation and does not take into consideration of the default aspect. Therefore catalogs based on this delivery can be created in the BMEcat2005 format.

2.1.4.3 eCl@ss advanced XML

The eCl@ss advanced XML format contains both the basic and advanced representation and is aware of the default aspect. To create a catalog according to the advanced XML delivery of the eCl@ss dictionary, the BMEcat 2005.1 format is required even when using the basic representation.

2.1.5 Summary: Representations and Deliveries

	Basic CSV	Basic XML	Advanced XML
Basic eCl@ss	BMEcat2005	BMEcat2005	BMEcat 2005.1
Advanced eCl@ss	n/a	n/a	BMEcat 2005.1

Table 3: eCl@ss representations, deliveries and required BMEcat versions

2.2 BMEcat

2.2.1 Transaction type

BMEcat defines three transaction types:

- T_NEW_CATALOG

- T_UPDATE_PRODUCTS

- T_UPDATE_PRICES

This document focuses on T_NEW_CATALOG.

2.2.2 Multivaluation

FEATURE_CONTENT.FT_VALENCY has two allowed values:

- multivalent: the defined property can have more than one value at a time

- univalent: the defined property can have only one value at a time

The eCl@ss standard is not defining if a property shall be used multivalent or univalent.

FEATURE.FVALUE_TYPE has three allowed values:

- choice: exactly one FVALUE is selected

- range: exactly two FVALUES are selected (denoting start and end value of the range), requires FT_DATATYPE of range-integer or range-numeric. In the advanced representation of the eCl@ss standard the property data type "level type" may be specified with min/max only which may be transported as a range.

- set: one or more FVALUEs are selected

FEATURE.FVALUE_TYPE has three allowed values:

- choice: exactly one FVALUE is selected

```
<FEATURE>
    <FTEMPLATE>
        <FT_ID>0173-1#02-BAA020#005</FT_ID>
        <FT_NAME>Höhe</FT_NAME>
    </FTEMPLATE>
    <FVALUE>35.000</FVALUE>
    <FVALUE>36.000</FVALUE>
    <FVALUE>37.000</FVALUE>
    <FUNIT>MMT</FUNIT>
    <FVALUE_TYPE>choice</FVALUE_TYPE>
    <FID>16</FID>
    <FPARENT_ID>12</FPARENT_ID>
</FEATURE>
```

- range: exactly two FVALUES are selected (denoting start and end value of the range), requires FT_DATATYPE of range-integer or range-numeric. In the advanced representation of the eCl@ss standard the property data type "level type" may be specified with min/max only which may be transported as a range.

```
<FEATURE>
    <FTEMPLATE>
        <FT_ID>0173-1#02-BAA020#005</FT_ID>
        <FT_NAME>Höhe</FT_NAME>
    </FTEMPLATE>
    <FVALUE>35.000</FVALUE>
    <FVALUE>37.000</FVALUE>
    <FUNIT>MMT</FUNIT>
    <FVALUE_TYPE>range</FVALUE_TYPE>
    <FID>16</FID>
    <FPARENT_ID>12</FPARENT_ID>
</FEATURE>
```

- set: one or more FVALUEs are selected

```
<FEATURE>
    <FTEMPLATE>
        <FT_ID>0173-1#02-BAA020#005</FT_ID>
        <FT_NAME>Höhe</FT_NAME>
    </FTEMPLATE>
    <FVALUE>35.000</FVALUE>
    <FVALUE>37.000</FVALUE>
    <FUNIT>MMT</FUNIT>
    <FVALUE_TYPE>set</FVALUE_TYPE>
    <FID>16</FID>
    <FPARENT_ID>12</FPARENT_ID>
</FEATURE>
```

2.2.3 Restrictions of data types

Using FT_FACET min/max length of strings, contained/non-contained upper/lower boundaries and number of digits can be controlled.

2.2.4 Ordering of properties

FEATURE.FORDER can (optionally) define in which order properties shall be showed in a target system. This must be in line with an order defined in a dictionary.

eCl@ss templates (see Section 2.4): Order of template SHOULD be kept and when template is "strict" it MUST be so.

2.2.5 Multiple classification systems

In a T_NEW_CATALOG a PRODUCT can have many PRODUCT_FEATURES. Each of these elements can be used to contain a reference to a different classification system.

2.2.6 Translatable vs. constant coded strings

<FVALUE> and <VALUE_IDREF> are of type multilingual string. Therefore it is always language specific.

If the eCl@ss standard defines a property as coded and if the desired value has a code assigned then the code shall be contained in the BMEcat in the following way:

<VALUE_IDREF>0173-1#07-AAA873#002</VALUE_IDREF>   // example for value "black"

If the value has not a code assigned the value shall be contained in the BMEcat.

<FVALUE>black</FVALUE>

When the eCl@ss data type of property is "string" (and not "translatable string"), it is sufficient to specify the value without "lang" as it can be derived from the dictionary that it is not translatable.

<FNAME> is used for properties with no assigned codes. For Non-eCl@ss properties an own code can be created according to ISO 6523.

2.3 Catalog and Dictionary change management

A catalog refers to a certain version of the dictionary. Over the course of time it might occur the situation that a receiving system accesses a different version of the sending system.

2.3.1 Dictionary of the catalog loading system newer than the catalog

Access to the earlier dictionary and TUF is recommended as properties may have been deprecated and valuation transformations may have been proposed together with dictionary updates. Still, starting with eCl@ss 6.2.1 to 7.0 it can be assumed that an old catalogue remains comprehensible with a new dictionary, except that:

- The code name / classification may have changed

- Some parts of the characterization may have become deprecated

2.3.2 Dictionary of the catalog loading system older than the catalog

Access to the later dictionary is recommended and required when new elements are contained in the catalog. However, according to eCl@ss dictionary change management rules, it can be taken for granted that wherever a catalog contains only later versions of known dictionary elements, their meaning will not have been restricted, i.e. when the catalog can be fully understood, given the old dictionary (not respecting version), it will not be invalid.

2.4 Catalog requirements

2.4.1 Templates

A template is a specification of the format and data requirements for a catalogue to meet the needs of a data recipient (e.g. a buyer).
Each template specifies a set of rules for the description of items that belong to a particular class, using references to concepts defined in a data dictionary, in order to meet the data requirements of a particular data recipient.
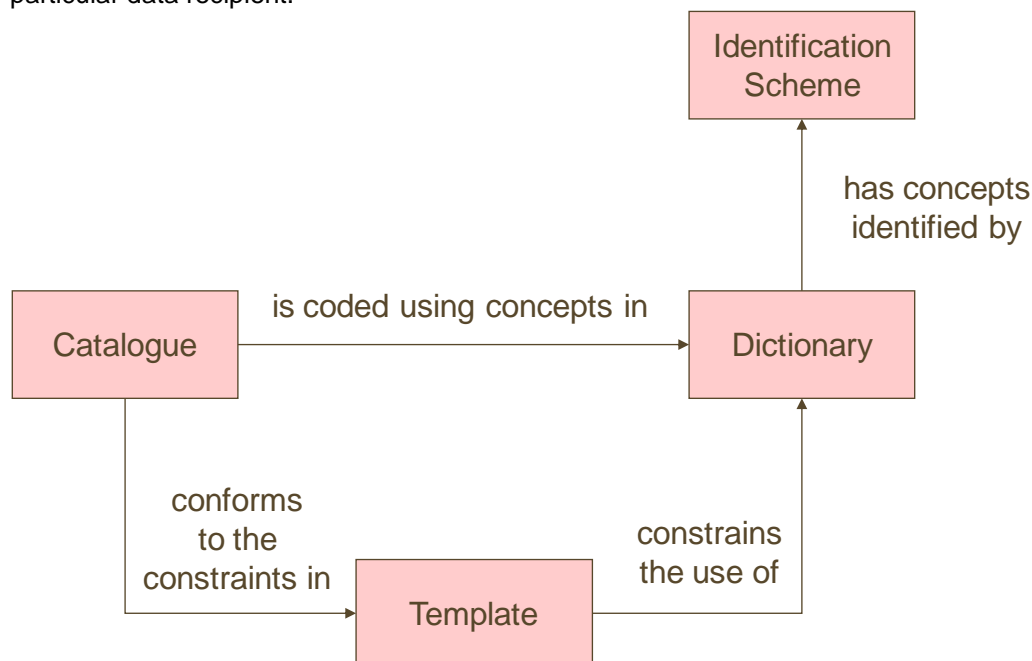
Figure 3 - Data architecture

Templates are constraining the use of the dictionary in catalogue data to meet the needs of a specific data recipient (e.g. buyer, group of buyers, etc.). The template is used to create a catalog according to the requirements on the catalog stated in the template. This means interpretation of a catalog must be possible without access to the template, since both the template and the catalog are based on the same dictionary.

2.4.2 Communication model

Templates are part of the communication layer (in contrast to the views that are part of the presentation).
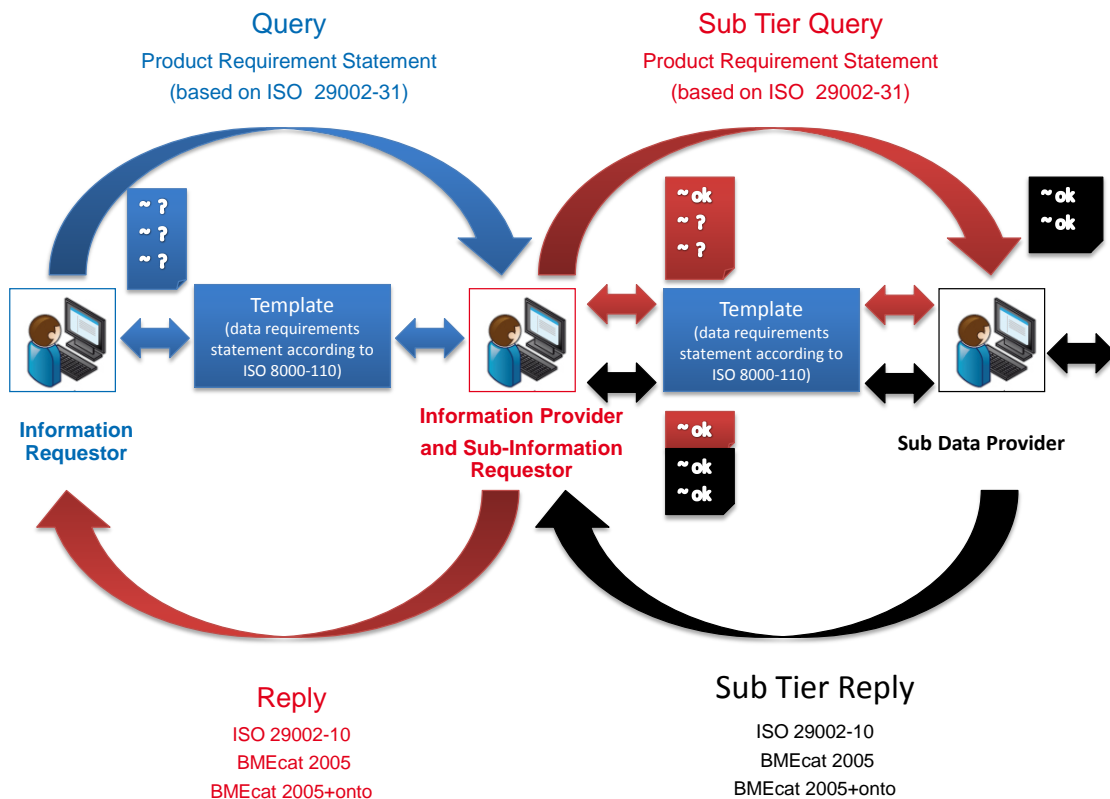
Figure 4 - communication model (assumption: every involved party has access to the dictionary)

Templates are developed and maintained by different actors than the maintainers of the dictionary. Dictionary providers can maintain a set of default templates as examples for the dictionary users.

Templates are modeling how the expected result (of a query) will look like.

eCl@ss uses an XML representation of templates defined within eCl@ss XML:

http://www.eclass.de/user/2011/templates/template.xsd

2.4.3 Relevant catalog information from template

The following information about requirements on the product description in a catalog can be obtained from templates:

- Visible properties

- Order or properties

- Naming of property in a particular language (override of dictionary)

- Value proposals for properties

- Generation rules for values of properties

- Change of default unit

- Mandatoryness and conditional mandatoryness of concerning the valuation of properties

- Flattening of structure

3 eCl@ss to BMEcat mapping

3.1 File naming scheme

File names should contain:

- Name of the catalog provider

- Name of the catalog recipient

- Time stamp

File names should not exceed 40 characters. Spaces should not be used in the file name.

3.2 Namespace

It is expected that the following start tag / name space is used for BMEcat2005:

```
<BMECAT version="2005" xmlns="http://www.eclass.eu/2011/bmecat/bmecat_2005">
```

It is expected that the following start tag / name space is used for BMEcat 2005.1:

```
<BMECAT version="2005" xmlns=" http://www.eclass.eu/2011/bmecat/ bmecat_2005onto">
```

3.3 REFERENCE_FEATURE_SYSTEM_NAME

The BMEcat specification gives a rule for formatting the content of <REFERENCE_FEATURE_SYSTEM_NAME>: The name of the feature system (in all capital letters) separated by a dash (minus sign) from the major and minor version that is given, separated by one dot. Moreover, the BMEcat spec has a list of known reference feature system names where the naming schema is explicitly mentioned.

eCl@ss 7 has to be written: ECLASS-7.0

3.4 Units of measure

BMEcat expects to find an UN/ECE code (String, up to three characters long) in the FUNIT element. See " Recommendation N°. 20 - Codes for Units of Measure Used in International Trade" under the following URL.

```
http://www.unece.org/cefact/recommendations/rec_index.htm
```

The transfer of the unit is a must. <FUNIT> is used for units with an assigned code and also for those without an assigned code. In the first case it is not possible to transfer a readable information.

3.5 Nationalization conventions for FVALUE

It is not clear from BMEcat specification what decimal separator character <T_NEW_CATALOG><PRODUCT><PRODUCT_FEATURES><FEATURE><FVALUE> should contain for numbers (e.g. valuation of real properties).

Example:

```
<FEATURE>
   <FTEMPLATE>
      <FT_ID>0173-1#02-BAA563#002</FT_ID>
      <FT_NAME>Innendurchmesser</FT_NAME>
```

```
    </FTEMPLATE>
    <FVALUE>108.00</FVALUE>
    <FUNIT>MMT</FUNIT>
</FEATURE>
```
Figure 5: XML Example FVALUE

Solution for valuation of eCl@ss properties: Always use dot notation (e.g. 108.00 and not 108,00)

3.6 MIME

3.6.1 MIME_SOURCE and MIME_ROOT

When using MIMEs there has to be defined where referenced files are expected. In BMEcat this is achieved by two elements: MIME_ROOT (in HEADER) which can specify a base path or an URI under which the relative paths specified in MIME_SOURCE start. The strings are concatenated by "/".

Example:

MIME_ROOT defines that all external files' relative paths start at http://www.example.com/img

MIME SORCE references "Charlie.jpg"

Referenced file is expected to be found under http://www.example.com/img/Charlie.jpg

Note: When no MIME_ROOT is specified there is no behavior defined by BMEcat specification. An application can try to assume that the MIME_ROOT is the directory where the catalog file is located, but this will not necessarily be interpreted alike in a target system.

3.6.2 MIME_DESCR and MIME_ALT

According to BMEcat specification the content of MIME_DESCR and MIME_ALT is only informational, no logic can be expected to be triggered

3.6.3 MIME_PURPOSE

In MIME_PURPOSE a catalog provider can specify whether the referenced file contains:

- data_sheet

- detail (enlarged image)

- icon

- logo

- normal (an image)

- safety_data_sheet

- thumbnail

- others (any other use)

Note: There is no guarantee that a target system interprets the content of MIME_PURPOSE in a special way.

Example:

```
<MIME_INFO>
   <MIME>
      <MIME_TYPE>image/jpeg</MIME_TYPE>
      <MIME_SOURCE>charlie.jpg</MIME_SOURCE>
      <MIME_DESCR>front view</MIME_DESCR>
      <MIME_ALT>picture charlie</MIME_ALT>
      <MIME_PURPOSE>normal</MIME_PURPOSE>
   </MIME>
   <MIME>
      <MIME_TYPE>application/pdf</MIME_TYPE>
      <MIME_SOURCE>charlie.pdf</MIME_SOURCE>
      <MIME_DESCR>description of the production process</MIME_DESCR>
      <MIME_ALT>pdf file about charlie</MIME_ALT>
      <MIME_PURPOSE>data_sheet</MIME_PURPOSE>   </MIME>
</MIME_INFO>
```

3.7 ARTICLE*

All elements named ARTICLE* are deprecated according to BMEcat specification. They have been replaced by reworked (extended) PRODUCT-elements. It is safe to assume for an application that the logic of an ARTICLE* element is taken over by the PRODUCT* element of the same name. An application producing new catalogs according to BMEcat2005 and BMEcat 2005.1 should make use of PRODUCT instead of ARTICLE.

3.8 HEADER Information

The elements from the table below show the mapping of elements included in catalogue header. The table shows the BMEcat2005 element, whether the element is mandatory, a German and English preferred name for display of the field and the kind of input required in an application.

The following elements are included in the header information.

BMEcat 2005 element (node complete name)	Mandatory/ Optional	proposed Display name DE	proposed Display name EN	Input by
GENERATOR_INFO	M	n/a	n/a	Application set Example:"eCatCreator 1.0"
CATALOG.DATETIME type=generation_date"	M	Erstelldatum	Generation date	Timestamp of generation date set by application (e.g. 2008-07-14)
CATALOG.LANGUAGE	M	Sprache	Language	User Default "deu" (according to ISO 639-2:1998)
CATALOG.CATALOG_ID	M	ID	ID	User entry
CATALOG.CATALOG_VERSION	M	Version	Version	User entry [major version.minor version] max.(xxx.yyy)
CATALOG.CATALOG_NAME	O	Name	Name	User entry
CATALOG.CATALOG_CURRENCY	M	Währung	Currency	User entry Default "EUR"
SUPPLIER.SUPPLIER_ID type:"duns"	O	DUNS Nr.	DUNS-ID	User entry http://www.dnbgermany.de/
SUPPLIER.NAME	M	Name des Lieferanten	Supplier Name	User
SUPPLIER.ADDRESS.EMAIL	O	Lieferant E-Mail	Supplier Email	User

Table 4: BMEcat 2005 Header information

Example :1 Header

```
<?xml version="1.0" encoding="utf-8" ?>
<BMECAT version="2005" xmlns="http://www.bmecat.org/bmecat/2005fd">
   <HEADER>
      <GENERATOR_INFO>eCl@ss DEMOcat Version 1.1</GENERATOR_INFO >
      <CATALOG>
         <LANGUAGE>deu</LANGUAGE>
         <CATALOG_ID>Hämmer 2008</CATALOG_ID>
         <CATALOG_VERSION>1.0</CATALOG_VERSION>
         <CATALOG_NAME>Hämmer 2008/1</CATALOG_NAME>
         <DATETIME type="generation_date">
            <DATE>2008-07-14</DATE>
         </DATETIME>
         <CURRENCY>EUR</CURRENCY>
      </CATALOG>
      <SUPPLIER>
         <SUPPLIER_NAME>Supplier AG</SUPPLIER_NAME>
         <ADDRESS type="supplier">
            <EMAIL>info@supplier-ag.de</EMAIL>
         </ADDRESS>
      </SUPPLIER>
   </HEADER>
…
</BMECAT>
```

Note: BMEcat should be UTF8 encoded by default.

3.9 Product Attributes

The elements from table below show the mapping of elements included in catalogue header. The table shows the BMEcat2005 element, whether the element is mandatory, a German and English preferred name for display of the field and the kind of input required in an application.

The following elements are included in the header information.

BMEcat 2005 element (node complete name)	BMEcat 2005 Mandatory/ Optional	proposed Display name DE	proposed Display name EN	Input by
SUPPLIER_PID	M	Lieferant-Artikelnr.	Supplier Article Number	User entry
DESCRIPTION_SHORT	M	Produktbeschreibung (kurz)	Product Description (short)	User entry
ORDER_UNIT	M	Bestelleinheit	Order Unit	User selection (list in Annex)
CONTENT_UNIT	O	Inhaltseinheit	Content Unit	User selection (list in Annex)
PRODUCT_PRICE@type	M	Preisinformationen	Price information	User selection List of entries
PRICE_AMOUNT	M	Preiswert	Price amount	User entry
TAX	M	Steuern	Tax factor	User entry

Table 5: BMEcat 2005 Product attributes

Example 2: Product Attributes

```
<T_NEW_CATALOG>
   <PRODUCT>
      <SUPPLIER_PID>444</SUPPLIER_PID>
      <PRODUCT_DETAILS>
         <DESCRIPTION_SHORT>Schlosserhammer 200g</DESCRIPTION_SHORT>
      </PRODUCT_DETAILS>
      <PRODUCT_ORDER_DETAILS>
         <ORDER_UNIT>C62</ORDER_UNIT>
         <CONTENT_UNIT>C62</CONTENT_UNIT>
      </PRODUCT_ORDER_DETAILS>
      <PRODUCT_PRICE_DETAILS>
         <PRODUCT_PRICE price_type="gros_list">
            <PRICE_AMOUNT>30.5</PRICE_AMOUNT>
            <TAX>0.19</TAX>
            <LOWER_BOUND>1</LOWER_BOUND>
         </PRODUCT_PRICE>
      </PRODUCT_PRICE_DETAILS>
      ...
   </PRODUCT>
...
</T_NEW_CATALOG>
```

3.10 Valuation of eCl@ss elements

This chapter describes the referencing and valuation of eCl@ss classes and properties

BMEcat 2005 element (node complete name)		eCl@ss element	Value	Input by
REFERENCE_FEATURE_SYSTEM_NAME		eCl@ss Version	ECLASS-7.0	Application
REFERENCE_FEATURE_GROUP_ID		eCl@ss classification no	27010390	Application
REFERENCE_FEATURE_GROUP_ID2 (type="flat")		ID of eCl@ss application class	0173-1#01-ACN104#004	Application
FTEMPLATE.FT_ID FTEMPLATE.FT_NAME		ID of the property Name of the property	0173-1#01-AAA001#001 Hersteller-Artikelnummer	Application
	FVALUE	Value of the property (for explicit values)	schwarz	User entry also multi-valuation possible
	OR			
	VALUE_IDREF	ID of the property value (for coded values)	0173-1#07-AAA873#002	User selection also multi-valuation possible
	FUNIT	Unit for FVALUE (if applicable for property/list of units)	Max. three letter ECE code, NO IRDI	Application/ User Selection

	FVALUE_DETAILS	More detailed description of a value	n/a	User selection

Table 6: Valuation of eCl@ss attributes

Remark 1: Name and ID are ALTERNATIVE entries in BMEcat 2005

There are multiple ways to identify a feature into the previously referenced classification or feature group system, recommended is:
By FTEMPLATE - In this case the definition of the property is specified (in place).

Remark 2
In case there is more than one application class classified under the same classification class in eCl@ss, specifying the in T_NEW_CATALOG.REFERENCE_FEATURE_GROUP_ID2 the IRDI of the actual application class is required.

Remark 3
Storing the ID of a value in FVALUE is not intended in BMEcat specification and it cannot be expected that an application will resolve an Identifier found in that field. If an identifier is to be placed, the field VALUE_IDREF has to be used.

Example 3: Valuation of eCl@ss elements

```
<REFERENCE_FEATURE_SYSTEM_NAME>ECLASS-7.0</REFERENCE_FEATURE_SYSTEM_NAME>
<REFERENCE_FEATURE_GROUP_ID>14050201</REFERENCE_FEATURE_GROUP_ID>
<REFERENCE_FEATURE_GROUP_ID2 type="flat"> 0173-1#01-ADK286#001
  </REFERENCE_FEATURE_GROUP_ID2>
<FEATURE>
  <FTEMPLATE>
    <FT_ID>0173-1#02-AAO663#001</FT_ID>
    <FT_NAME>GTIN</FT_NAME>
  </FTEMPLATE>
  <FVALUE>00012345600012</FVALUE >
  <FID>1</FID>
  <FPARENT_ID>-1</FPARENT_ID>
  <!-- … // -->
</FEATURE>
```

Remark 4
It is necessary to define the complete structure of attributes in the BMEcat. Please see the examples for blocks, aspects, cardinality, polymorphism, and level type.

Remark 5
eCl@ss 7.0 supports the level type "min" and "max". The first value is always the "min" value, the second value is the "max" value.

Example 4: Level type

```
<FEATURE>
  <FVALUE>2.5</FVALUE >            // min-value
  <FVALUE>4.0</FVALUE >            // max-value
  <FVALUE_TYPE>range</FVALUE_TYPE>
</FEATURE>
```

3.11 PRODUCT DETAILS

3.11.1 KEYWORD

Under PRODUCT_DETAILS there can be given KEYWORDs for items. An application should allow to utilize the keywords in search for items.

3.11.2 DESCRIPTION_LONG

This field may contain text or preformatted (with < and > escaped into entities as > and <) HTML describing the item.

3.11.3 SPECIAL_TREATMENT_CLASS

The element is BMEcat only, no mapping to eCl@ss 7.0 possible.

3.11.4 SEGMENT

Segment is alternative classification to eCl@ss.

3.11.5 Mapping of eCl@ss BML to BMEcat 2005 PRODUCT_DETAILS elements

The table below shows a mapping of elements in eCl@ss 7.0 Basic List of Properties (BML) to the corresponding BMEcat 2005 elements.
The objective of the mapping is to suggest value equivalence and implies identical values to be put in BMEcat attributes and eCl@ss property valuation.

BMEcat 2005 element (node complete name)	Mandatory / Optional	proposed Display name DE	eCl@ss Rel. 7.0 property ID	Input by
DESCRIPTION_SHORT	M	Artikelbezeichnung	0173-1#02-AAP805#001	User entry See Remark 1
DESCRIPTION_LONG	O	Langbeschreibung	n/a	User entry
INTERNATIONAL_PID type=GTIN	O	GTIN	0173-1#02-AAO663#001	User entry Supersedes type=EAN and EAN field
MANUFACTURER_NAME	O	Hersteller-Name	0173-1#02-AAO677#001	User entry
MANUFACTURER_PID	O	Hersteller-Artikelnummer	0173-1#02-AAO676#001	User entry
SUPPLIER_PID	O	Lieferanten-Artikelnummer	0173-1#02-AAO736#001	User entry
MANUFACTURER_TYPE_ DESCR	O	Produkt-Typbezeichnung	0173-1-02- AAO847#001	Partial match, see remark 3

Table 7: Mapping eCl@ss BML – BMEcat 2005 elements

Remark 1:
According to BMEcat DESCRIPTION_SHORT may be up to 150 characters long, but **shall be unique within the 40 first characters** (to be usable with systems like SAP). It should not contain abbreviated terms and have a clear and expressive language. Abbreviations of organizations and standards (e.g. DIN A4) may be used. DESCRIPTION_SHORT is not fully equal to 0173-1#02-AAP805-001 (Artikelbezeichnung), but the value may be taken over as a shortcut without violating the BMEcat specification.

Note 1: DESCRIPTION_SHORT should not simply be set to the preferred name of the class according to which the product is described due to the desired uniquity

Note 2: DESCRIPTION_LONG may contain a longer (up to 64k characters) textual description of the product (containing even HTML with < and > written as entities for XML compatibility).

Remark 2:

The mapping in table 7 shall be used in this way until eCl@ss delivers a better mapping with one of the following releases.

Remark 3:

DESCRIPTION_SHORT can be interpreted as a "preferred name" for the product
DESCRIPTION_LONG can be interpreted as a "definition" for the product

Remark 4:

PRODUCT_TYPE in BMEcat allows to specify from a list of values the generic type of the product (like: part of a bundle, a contract, …) whereas the eCl@ss property 0173-1-02- AAO847#001 allows to specify a string that can be an abbreviated name of the product or a short description of its purpose. Therefore MANUFACTURER_TYPE_
DESCR, which contains a name supplied by the manufacturer that may be better known than the one supplied in DESCRIPTION_SHORT, seems like the better match here.

Remark 5:

BMEcat and eCl@ss allow to transport the same information in different elements. See table above. If the value of an identical eCl@ss feature differs from a BMEcat feature, the BMEcat value is effective.

Note:

The eCl@ss property code is fix. Only the version information can be changed with following eCl@ss releases.

Annex

A - UN/ECE Recommendation 20 for Valid / Used Order and Content-units

For specifications on order- and content units see UN/ECE Recommendation 20 at
http://www.unece.org/